Technical Cybersecurity

objdump

GNU Binutils

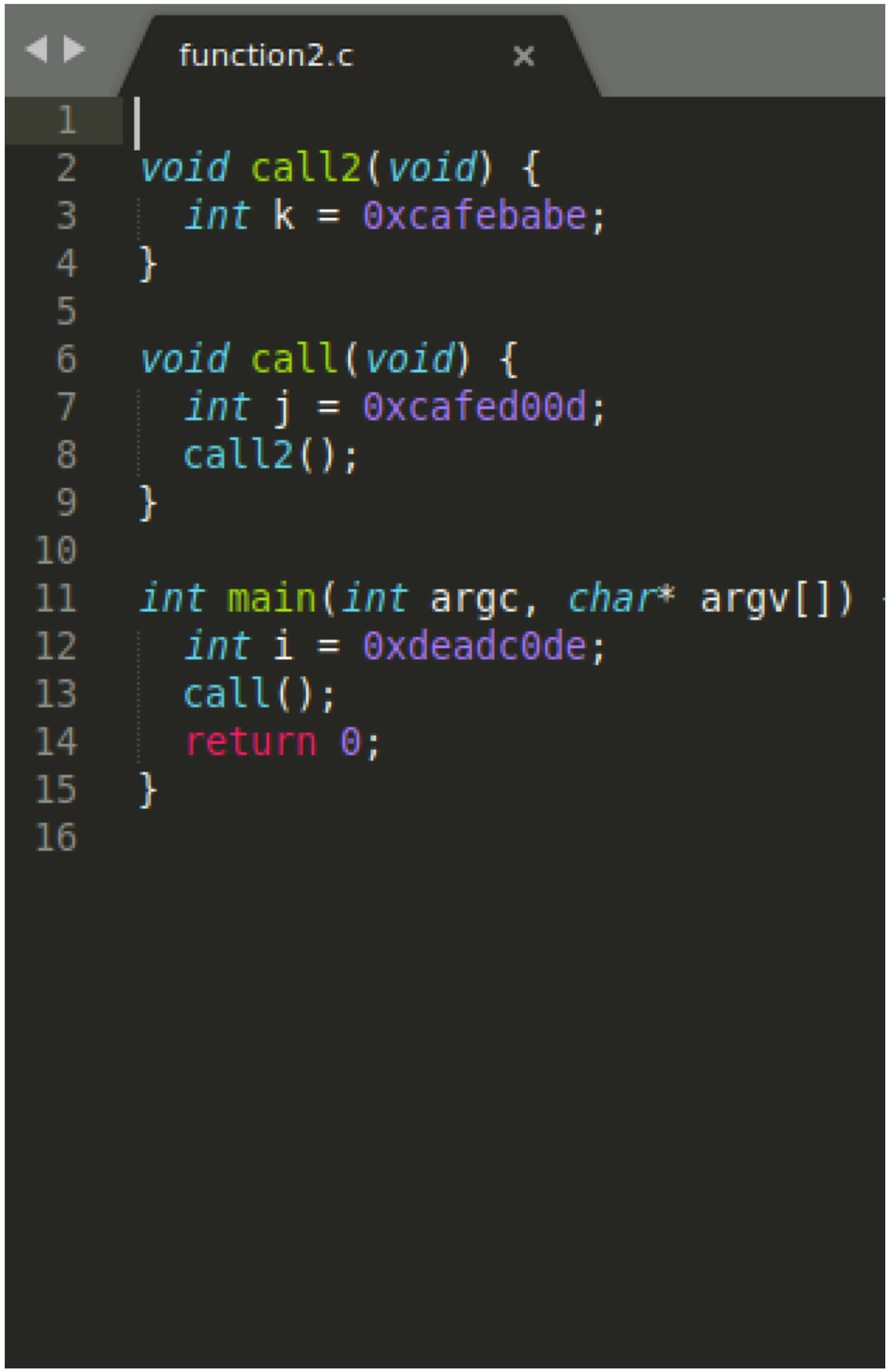
Contains lots of useful tools

- objdump
- readelf
- strings
- nm
- strip
- …and many more

Sample C Program

WHAT IS THIS?

- A simple program
- Calls two functions
- We'll use to example stack management, registers, etc.

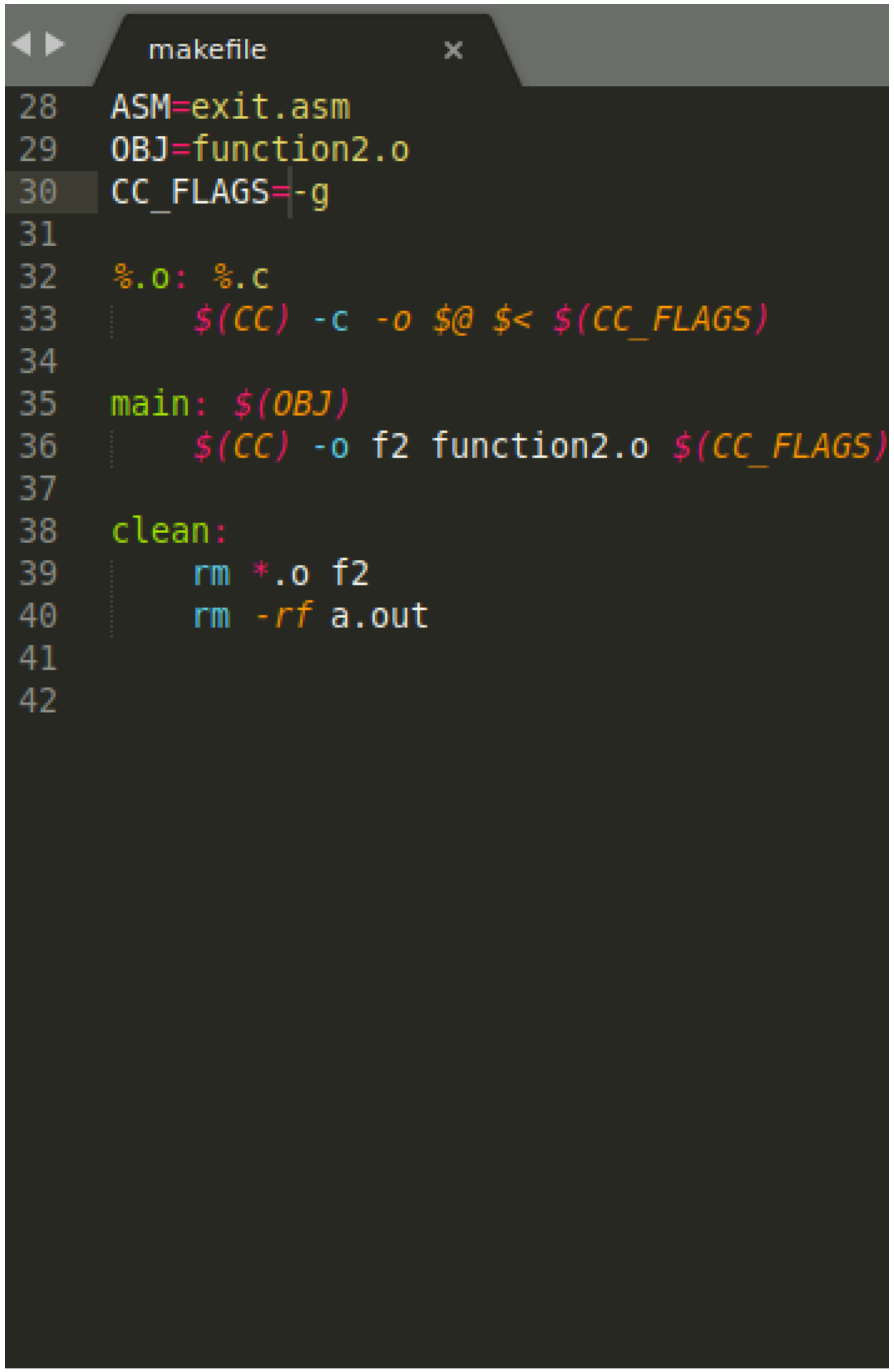


```c

void call2(void) {
  int k = 0xcafebabe;
}

void call(void) {
  int j = 0xcafed00d;
  call2();
}

int main(int argc, char* argv[])
  int i = 0xdeadc0de;
  call();
  return 0;
}

```

Sample makefile

MAKEFILES USE TABS

- Remember, you need to *tab* in from target names
- This will compile the function2.c file
- We'll add flags
- **$ make**



makefile

```
ASM=exit.asm
OBJ=function2.o
CC_FLAGS=-g

%.o: %.c
	$(CC) -c -o $@ $< $(CC_FLAGS)

main: $(OBJ)
	$(CC) -o f2 function2.o $(CC_FLAGS)

clean:
	rm *.o f2
	rm -rf a.out
```

objdump

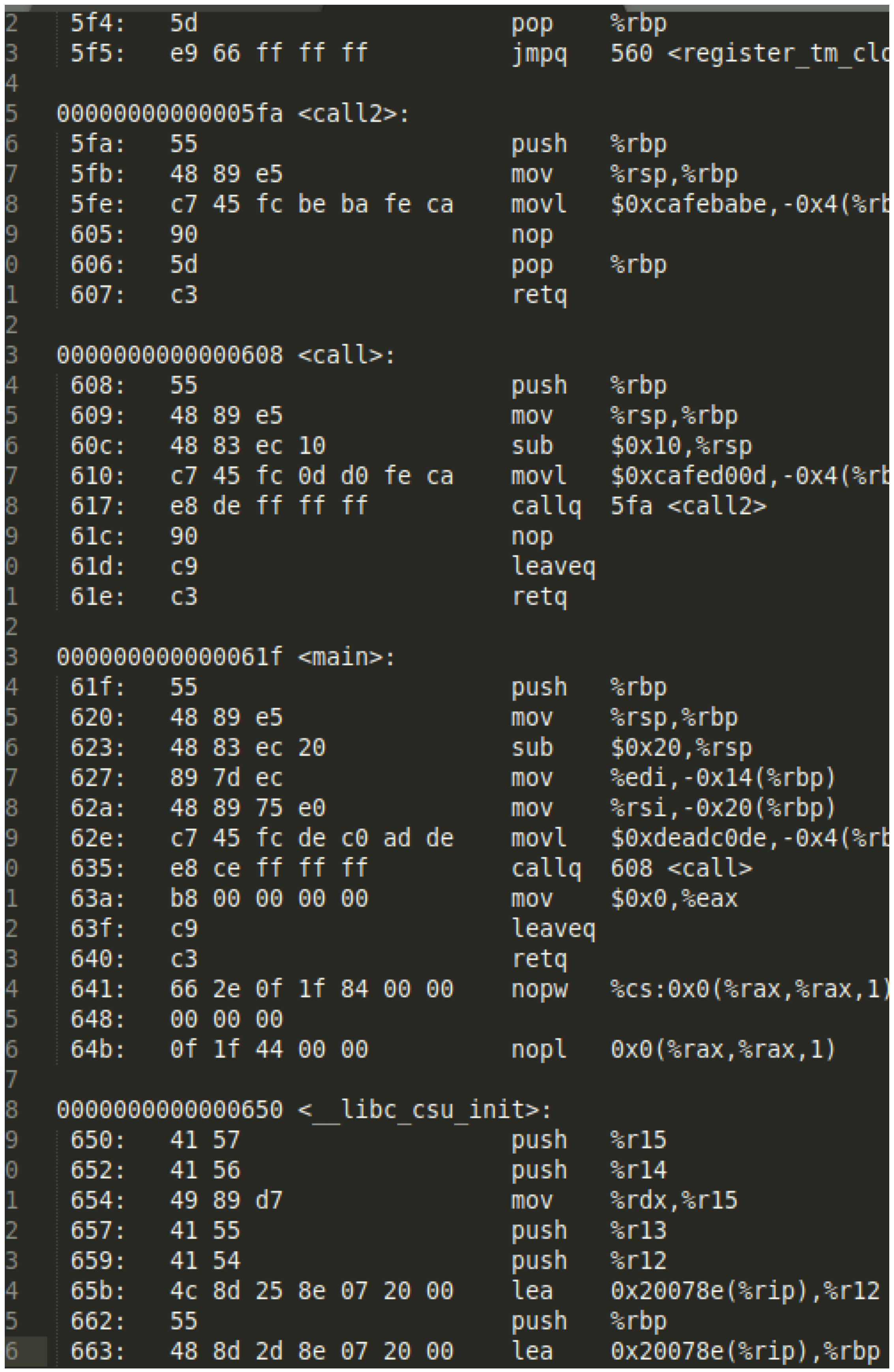
The executable

- **$ objdump**

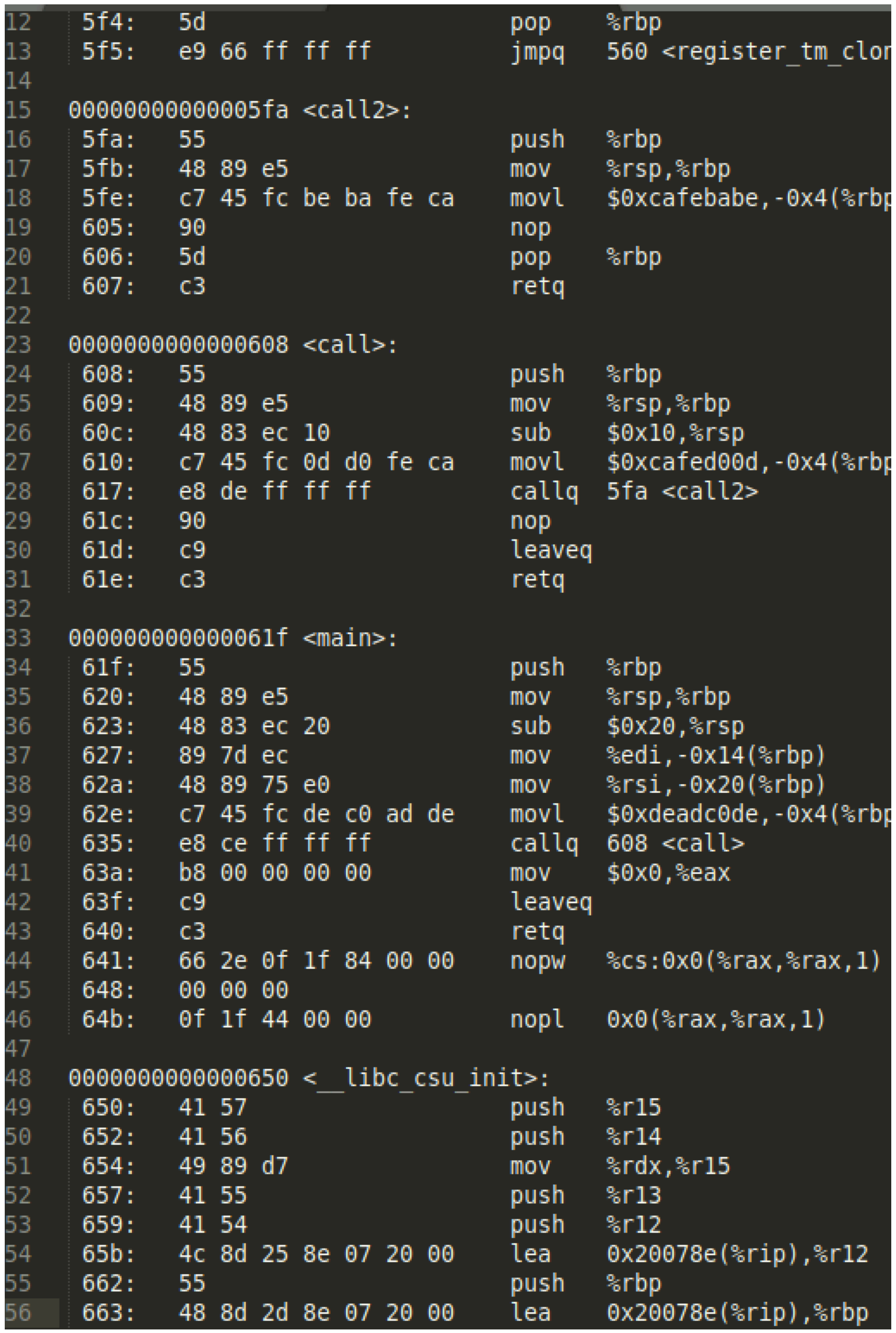
Disassembly

- **$ objdump -d f2 > f2.dis**

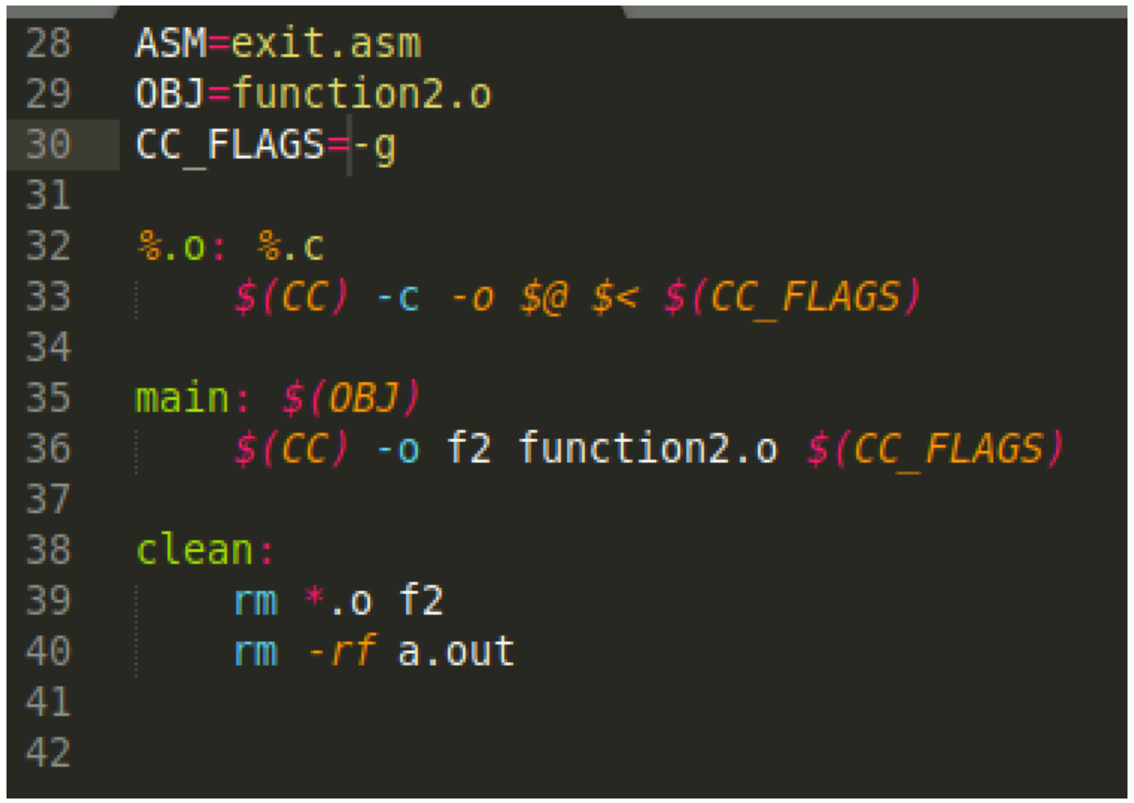
See the hexspeak?



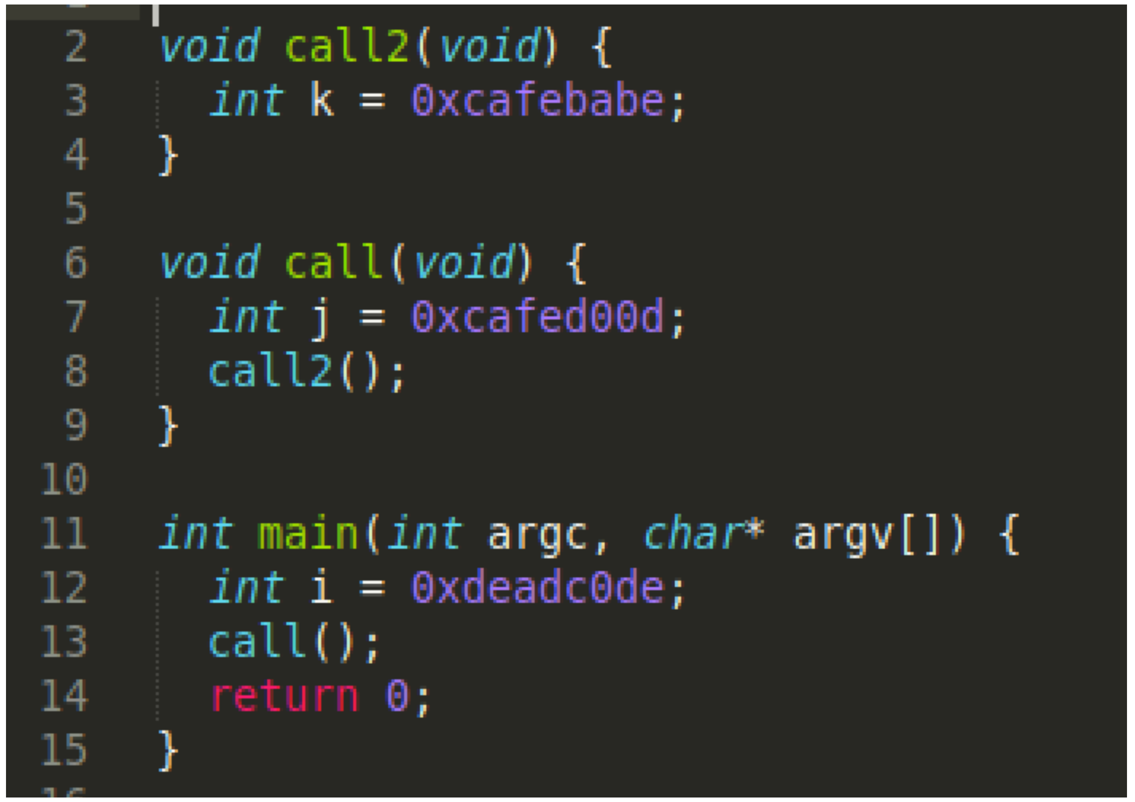
```
 5f4:	5d                   	pop    %rbp
 5f5:	e9 66 ff ff ff       	jmpq   560 <register_tm_clo

00000000000005fa <call2>:
 5fa:	55                   	push   %rbp
 5fb:	48 89 e5             	mov    %rsp,%rbp
 5fe:	c7 45 fc be ba fe ca 	movl   $0xcafebabe,-0x4(%rb
 605:	90                   	nop
 606:	5d                   	pop    %rbp
 607:	c3                   	retq

0000000000000608 <call>:
 608:	55                   	push   %rbp
 609:	48 89 e5             	mov    %rsp,%rbp
 60c:	48 83 ec 10          	sub    $0x10,%rsp
 610:	c7 45 fc 0d d0 fe ca 	movl   $0xcafed00d,-0x4(%rb
 617:	e8 de ff ff ff       	callq  5fa <call2>
 61c:	90                   	nop
 61d:	c9                   	leaveq
 61e:	c3                   	retq

000000000000061f <main>:
 61f:	55                   	push   %rbp
 620:	48 89 e5             	mov    %rsp,%rbp
 623:	48 83 ec 20          	sub    $0x20,%rsp
 627:	89 7d ec             	mov    %edi,-0x14(%rbp)
 62a:	48 89 75 e0          	mov    %rsi,-0x20(%rbp)
 62e:	c7 45 fc de c0 ad de 	movl   $0xdeadc0de,-0x4(%rb
 635:	e8 ce ff ff ff       	callq  608 <call>
 63a:	b8 00 00 00 00       	mov    $0x0,%eax
 63f:	c9                   	leaveq
 640:	c3                   	retq
 641:	66 2e 0f 1f 84 00 00 	nopw   %cs:0x0(%rax,%rax,1)
 648:	00 00 00
 64b:	0f 1f 44 00 00       	nopl   0x0(%rax,%rax,1)

0000000000000650 <__libc_csu_init>:
 650:	41 57                	push   %r15
 652:	41 56                	push   %r14
 654:	49 89 d7             	mov    %rdx,%r15
 657:	41 55                	push   %r13
 659:	41 54                	push   %r12
 65b:	4c 8d 25 8e 07 20 00 	lea    0x20078e(%rip),%r12
 662:	55                   	push   %rbp
 663:	48 8d 2d 8e 07 20 00 	lea    0x20078e(%rip),%rbp
```



```
 5f4:   5d                      pop    %rbp
 5f5:   e9 66 ff ff ff          jmpq   560 <register_tm_clon

00000000000005fa <call2>:
 5fa:   55                      push   %rbp
 5fb:   48 89 e5                mov    %rsp,%rbp
 5fe:   c7 45 fc be ba fe ca    movl   $0xcafebabe,-0x4(%rbp
 605:   90                      nop
 606:   5d                      pop    %rbp
 607:   c3                      retq

0000000000000608 <call>:
 608:   55                      push   %rbp
 609:   48 89 e5                mov    %rsp,%rbp
 60c:   48 83 ec 10             sub    $0x10,%rsp
 610:   c7 45 fc 0d d0 fe ca    movl   $0xcafed00d,-0x4(%rbp
 617:   e8 de ff ff ff          callq  5fa <call2>
 61c:   90                      nop
 61d:   c9                      leaveq
 61e:   c3                      retq

000000000000061f <main>:
 61f:   55                      push   %rbp
 620:   48 89 e5                mov    %rsp,%rbp
 623:   48 83 ec 20             sub    $0x20,%rsp
 627:   89 7d ec                mov    %edi,-0x14(%rbp)
 62a:   48 89 75 e0             mov    %rsi,-0x20(%rbp)
 62e:   c7 45 fc de c0 ad de    movl   $0xdeadc0de,-0x4(%rbp
 635:   e8 ce ff ff ff          callq  608 <call>
 63a:   b8 00 00 00 00          mov    $0x0,%eax
 63f:   c9                      leaveq
 640:   c3                      retq
 641:   66 2e 0f 1f 84 00 00    nopw   %cs:0x0(%rax,%rax,1)
 648:   00 00 00
 64b:   0f 1f 44 00 00          nopl   0x0(%rax,%rax,1)

0000000000000650 <__libc_csu_init>:
 650:   41 57                   push   %r15
 652:   41 56                   push   %r14
 654:   49 89 d7                mov    %rdx,%r15
 657:   41 55                   push   %r13
 659:   41 54                   push   %r12
 65b:   4c 8d 25 8e 07 20 00    lea    0x20078e(%rip),%r12
 662:   55                      push   %rbp
 663:   48 8d 2d 8e 07 20 00    lea    0x20078e(%rip),%rbp
```



```
ASM=exit.asm
OBJ=function2.o
CC_FLAGS=-g

%.o: %.c
	$(CC) -c -o $@ $< $(CC_FLAGS)

main: $(OBJ)
	$(CC) -o f2 function2.o $(CC_FLAGS)

clean:
	rm *.o f2
	rm -rf a.out
```



```
void call2(void) {
  int k = 0xcafebabe;
}

void call(void) {
  int j = 0xcafed00d;
  call2();
}

int main(int argc, char* argv[]) {
  int i = 0xdeadc0de;
  call();
  return 0;
}
```

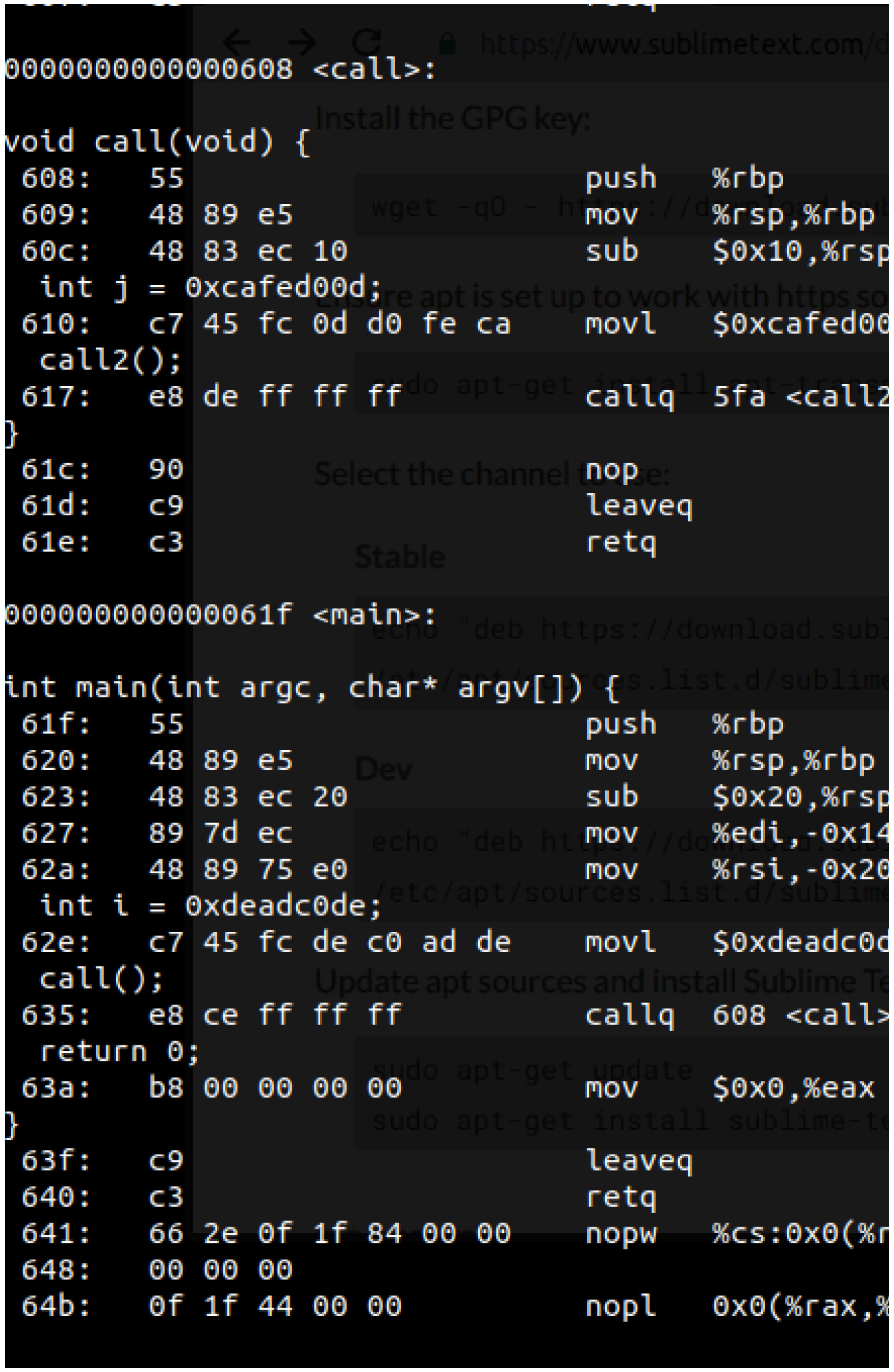
Other options

MIX SOURCE & ASM

- **$ objdump -S f2 | less**

LOAD FLAGS

- **$ objdump -f f2**



```
0000000000000608 <call>:

void call(void) {
 608:   55                      push   %rbp
 609:   48 89 e5                mov    %rsp,%rbp
 60c:   48 83 ec 10             sub    $0x10,%rsp
  int j = 0xcafed00d;
 610:   c7 45 fc 0d d0 fe ca    movl   $0xcafed00
  call2();
 617:   e8 de ff ff ff          callq  5fa <call2
}
 61c:   90                      nop
 61d:   c9                      leaveq
 61e:   c3                      retq

000000000000061f <main>:

int main(int argc, char* argv[]) {
 61f:   55                      push   %rbp
 620:   48 89 e5                mov    %rsp,%rbp
 623:   48 83 ec 20             sub    $0x20,%rsp
 627:   89 7d ec                mov    %edi,-0x14
 62a:   48 89 75 e0             mov    %rsi,-0x20
  int i = 0xdeadc0de;
 62e:   c7 45 fc de c0 ad de    movl   $0xdeadc0d
  call();
 635:   e8 ce ff ff ff          callq  608 <call>
  return 0;
 63a:   b8 00 00 00 00          mov    $0x0,%eax
}
 63f:   c9                      leaveq
 640:   c3                      retq
 641:   66 2e 0f 1f 84 00 00    nopw   %cs:0x0(%r
 648:   00 00 00
 64b:   0f 1f 44 00 00          nopl   0x0(%rax,%
```

More Binutils coming up!